



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

APPLICATION OF : Peterson et al.  
SERIAL NUMBER : 09/886,302  
FILED : June 21, 2001  
FOR : CONDITIONING THE EXECUTION OF AN  
EXECUTABLE PROGRAM UPON  
SATISFACTION OF CRITERIA  
EXAMINER : Shin Hon Chen  
Art Group : 2131

**BRIEF ON APPEAL**

**1. REAL PARTY IN INTEREST**

The application is assigned to Lockheed Martin Corporation, and was recorded on June 21, 2001 at Reel 011956, frame 0520.

**2. RELATED APPEALS AND INTERFERENCES**

None

**3. STATUS OF CLAIMS**

The application was originally filed with 10 claims, of which claims 1 and 9 were independent. In a first Office Action, all claims were rejected. In response, independent claims 1 was cancelled, claims 2 and 10 were amended to independent form, and changes to the dependency of other claims were made. A final Office Action continued the rejection of claims 2-8 and 10.

Appeal is taken from the rejection of claims 2-8 and 10.

07/01/2005 HLE333 00000051 502061 09886302

01 FC:1402 500.00 DA

**4. STATUS OF AMENDMENTS**

No amendments after final rejection are made.

**5. SUMMARY OF THE INVENTION**

The invention relates to a method for tending to reduce the possibility of virus infection of an intranet which communicates by way of a virtual private network (VPN) with a remote computer which is used for other purposes.

The remote computer is subject to the possibility of

07/01/2005 HLE333 00000005 502061 09886302

01 FC:1401 500.00 DA

Void date: 07/01/2005 HLE333

07/01/2005 HLE333 00000005 502061 09886302

01 FC:1401 500.00 CR

42 infection, which infection might be communicated to the  
intranet through the VPN (page 5, line 7 to page 7, line 6).

According to an aspect of the invention, the  
underlying VPN-generating program (or other executable  
program) is appended to, or ``encapsulated'' in an  
executable policy enforcement agent including a header, an  
48 execution portion, and a data portion, to thereby form a  
combined program (page 7, line 8 to page 8, line 7).  
Another view of the encapsulation is that of substitution of  
the header of the policy enforcement agent for the header of  
the underlying application. The purpose of the  
encapsulation is to reduce the possibility of direct  
54 invocation of the underlying program and thereby avoiding  
the policy. In the context of the VPN-generating program,  
this corresponds to preventing execution until an antivirus  
program has executed. When the underlying program is to be  
invoked, the combined program is invoked (page 9, lines 25-  
30), which in turn invokes the policy enforcement agent.  
60 The policy enforcement agent requires that the policy be  
fulfilled, as for example by running an antivirus program,  
before allowing execution of the underlying program, such as  
the VPN-generating software (page 9, line 30 to page 10,  
line 8).

An advantage of the encapsulated executable  
66 program according to an aspect of the invention is that it  
can be moved from one computer to another, without requiring  
any changes to the new or receiving computer, and the  
encapsulated program will, in the new computer, have the  
same effect as in the old computer.

72 6 ISSUES

1. Claims 2 and 10 are patentable in a 35 U.S.C.  
§102(e) sense over the cited O'Brien et al. reference.

2. Claims 3-8 and 10 are patentable in a 35 U.S.C.  
§103(a) sense.

78 7. GROUPING OF CLAIMS

Claims 2, 3-8 and 10 stand or fall together.

## 8. ARGUMENT

### 8A. The References

84 The O'Brien reference (U.S. 6,658,571) is a  
computer security system, in which access to computer  
resources such as processing units, ROM, RAM, or busses are  
selectively withheld from operating programs (column 3,  
lines 2-25, 39-49) by security modules if they execute  
malicious software. Note that the security modules (105)  
can be loaded within kernel 102 while computer system 100 is  
90 running (column 3, lines 56-64) to provide the security  
function as to an executing underlying programs 107. In  
short, O'Brien et al. selectively withhold computer  
resources from currently running underlying programs in  
accordance with their security programming.

It should be noted that O'Brien does not prevent  
96 an uninfected application program, operating in an infected  
computing environment, from becoming infected during the  
period of its operation prior to the execution of a  
prohibited task.

### 8B. Anticipation

102 There is a salient difference between the claimed  
arrangement and the O'Brien arrangement. Security in  
O'Brien et al. depends upon the security modules 105 of  
FIGURE 1 of O'Brien, which are preloaded into kernel 102  
(column 3, lines 55-56), apart from applications 107, which  
execute in user space (column 3, lines 29-37). Thus, the  
108 simple transfer of an application, such as 107 of O'Brien et  
al., to a new computer, will not transfer the security  
aspects as in the arrangement of the claimed invention.  
Instead, other measures must be taken, such as additionally  
transferring the security module. As to any particular  
application, the security in O'Brien is provided by software  
114 preloaded into the computer, rather than by the encapsulated

program or application itself. These differences arise from the recitations of the claims, as set forth below.

Claims 2 and 10 are rejected as anticipated by O'Brien et al. Claim 2 as amended recites inter alia

120 "substituting said combined program for said executable application, so that said policy enforcement agent executes instead of said executable application program when said executable application is invoked; and

126 one of (a) satisfying said conditions of said control module, whereby said executable application executes, and (b) not satisfying said conditions, whereby said executable application does not execute;

132 wherein said software executable policy enforcement agent includes a header component, and said substituting step includes the step of amending said header component of said policy enforcement agent portion of said combined program to match the characteristics of said combined program."

138 It does not appear that the O'Brien arrangement meets any of these limitations of claim 2. More particularly, it appears that the O'Brien software program(s) execute(s) independently of the security modules, as the security modules have nothing on which to act unless the underlying programs make calls for system resources, which can only occur if the  
144 underlying programs are already running. Thus, the security modules do not alternatively

150       "(a) satisfy[ing] said conditions of said control  
module, whereby said executable application  
executes, and (b) not satisfying said conditions,  
whereby said executable application does not  
execute"

as recited in claim 2

Further, Examiner states (Final Rejection, page 3)

156       "O'Brien further discloses wherein said software  
executable policy enforcement agent includes a  
header component, and said substituting step  
includes the step of amending said header  
component of said policy enforcement agent portion  
of said combined program to match the  
characteristics of said combined program (O'Brien:  
column 2 lines 12-38 . . . .")

162       Examiner is clearly wrong in this regard, as O'Brien  
makes no mention whatever of "header" or  
"substitution." Thus, each and every element of  
claim 2 is not found in O'Brien, and the requirements  
of anticipation are not met. In the absence of a  
showing in O'Brien of each and every claimed element of  
claim 2, there can be no anticipation.

168       Claim 2 is clearly patentable in a 35 U.S.C.  
§102(e) sense over O'Brien. Since Examiner indicates  
that claim 10 has the same scope as claim 2, claim 10  
is also patentable.

#### 8C. Obviousness

174       Examiner premises the 35 U.S.C. §103(a) rejection  
of dependent claims 3-8 on the same principal reference  
(O'Brien et al.) as that used for the anticipation

rejection. As argued above, independent claims 2 and 10 are patentable in an anticipation sense. Thus, dependent claims 2-8 depend from patentable parent claim 2, and they are patentable therewith.

180

#### 9. AUTHORITIES RELIED UPON

For the proposition that there must be identity of each and every element of the claimed invention and the reference in order to find anticipation, appellant relies upon one or more of RCA Corp. v Applied Digital Data

186

Systems, Inc. 221 USPQ 385, 388 (Fed. Cir. 1984); Kalman v Kimberly-Clark Corp., 218 USPQ 781, 789 (Fed. Cir. 1983); Orthokinetics, Inc. v Safety Travel Chairs, Inc., 1 U.S.P.Q. 2<sup>d</sup> 1081, 1087 (Fed. Cir. 1986); Hybritech, Inc. v Monoclonal Antibodies, Inc., 231 USPQ 81, 90 (Fed. Cir. 1986); Carella v Starlight Archery & Pro Line Co., 231 USPQ 644, 646 (Fed. Cir. 1986).

192

For the proposition that a dependent claim is non-obvious if it depends from a patentable claim, appellants rely on In re Fine, 5 USPQ2d 1596, 1600 (Fed. Cir. 1988), citing Hartness Int'l v Simplimatic Eng'g Co., 2 USPQ2d 1826, 1831; In re Abele, 214 USPQ 682, 689 (CCPA 1982)

198

#### 10. CONCLUSION

Claims 2 and 10 are patentable in an anticipation sense over Examiner's suggested anticipatory reference.

Examiner's rejection of claims 2 and 10 should be reversed, together with dependent claims 2 to 8. Reversal of Examiner's rejection is requested.

204

210 11. Please charge the fee for the appeal brief to 50-  
2061.

216

Respectfully Submitted



William H. Meise  
Reg. No. 27,574

222 June 28, 2005  
IN TRIPLICATE

## CLAIMS

1. (Cancelled) A security method for controlling use of an executable application, said method comprising the steps of:

228           procuring a software executable policy enforcement agent which, when invoked, imposes one or more conditions on successful execution, and which, when successfully executed, invokes execution of said executable application;

                  encapsulating said executable application with  
234   said policy enforcement agent without changing said executable application, to thereby produce a combined program;

                  substituting said combined program for said executable application, so that said policy enforcement agent executes instead of said executable application  
240   program when said executable application is invoked; and

                  one of (a) satisfying said conditions of said control module, whereby said executable application executes, and (b) not satisfying said conditions, whereby said executable application does not execute.

246           2. (Previously Amended) A security method for controlling use of an executable application, said method comprising the steps of:

                  procuring a software executable policy enforcement agent which, when invoked, imposes one or more conditions on successful execution, and which, when  
252   successfully executed, invokes execution of said executable application;



encapsulating said executable application with said policy enforcement agent without changing said executable application, to thereby produce a combined program;

258 substituting said combined program for said executable application, so that said policy enforcement agent executes instead of said executable application program when said executable application is invoked; and  
one of (a) satisfying said conditions of said control module, whereby said executable application  
264 executes, and (b) not satisfying said conditions, whereby said executable application does not execute;

wherein said software executable policy enforcement agent includes a header component, and said substituting step includes the step of amending said header component of said policy enforcement agent portion of said  
270 combined program to match the characteristics of said combined program.

3. (Previously Amended) A method according to claim 2, wherein said executable application includes a VPN-tunnel-generating application, and said step of  
276 satisfying said conditions includes the step of running an antivirus program.

4. (Previously Amended) A method according to claim 2, wherein said executable application includes a VPN-tunnel-generating application, and said step of  
282 satisfying said conditions includes the step of running an antivirus program having an acceptable update status.

5. (Previously Amended) A method according to claim 2, wherein said step of satisfying said conditions includes the step of running a personal firewall program.

288

6. (Previously Amended) A method according to claim 2, wherein said executable application accepts verification information in a format other than a digital certificate, and said step of satisfying said conditions includes the step of accepting a digital certificate.

294

7. (Original) A method according to claim 6, wherein said step of accepting a digital certificate includes the step of accepting an X.509 based digital certificate.

300

8. (Original) A method according to claim 6, further comprising the step of translating at least some information from said digital certificate into a form recognizable by said executable application.

306

9. (Cancelled) A method for policy enforcement in relation to an executable application, said method comprising the steps of:

312

procuring a software control element which is identifiable to a host operating system as an executable program and which includes an execution component for executing said executable application, and which also contains a set of conditions which must be met in order to invoke said executable application;

combining said software control element with said executable application, to form a combined program;

substituting said combined program for said executable application;

318           commanding execution of said combined program, to  
thereby execute said software control element, whereupon  
said execution component is invoked if said conditions are  
met, and said executable application executes.

10. (Previously Amended) A method for policy  
324 enforcement in relation to an executable application, said  
method comprising the steps of:

          procuring a software control element which is  
identifiable to a host operating system as an executable  
program and which includes an execution component for  
executing said executable application, and which also  
330 contains a set of conditions which must be met in order to  
invoke said executable application;

          combining said software control element with said  
executable application, to form a combined program;

          substituting said combined program for said  
executable application;

336           commanding execution of said combined program, to  
thereby execute said software control element, whereupon  
said execution component is invoked if said conditions are  
met, and said executable application executes;

          wherein software control element includes a  
header identifying the locations of executable and data  
342 portions of said control element, and said step of  
combining said software control element with said  
executable application includes the steps of:

appending said executable application to said software control element in a location identified by said software control element as a data location; and

348 updating said header of said software control module to correspond with the characteristics of said combined program.